Barrier Proof

March 16, 2016

This document gives the Iris version of the specification and proof of the *barrier* originally presented in [1].

1 Code and spec

We aim to verify the following piece of code:

$$\begin{aligned} \texttt{newbarrier} &\triangleq \lambda\_.\, \texttt{ref}(0) \\ \texttt{signal} &\triangleq \lambda x.\, x \leftarrow 1 \\ \texttt{wait} &\triangleq \texttt{rec}\ \mathit{wait}(x :) = \texttt{if}\ !x = 1\ \texttt{then}\ ()\ \texttt{else}\ \mathit{wait}() \end{aligned}$$

with respect to the following specification: There exists recv, send such that

ALLOC
$\{\mathsf{True}\}\ \texttt{newbarrier}()\ \{\ell.\, \mathsf{recv}(\ell, P) * \mathsf{send}(\ell, P)\}_\top$

SIGNAL
$\{\mathsf{send}(\ell, P) * P\}\ \texttt{signal}(\ell)\ \{\mathsf{True}\}_\top$

WAIT
$\{\mathsf{recv}(\ell, P)\}\ \texttt{wait}(\ell)\ \{P\}_\top$

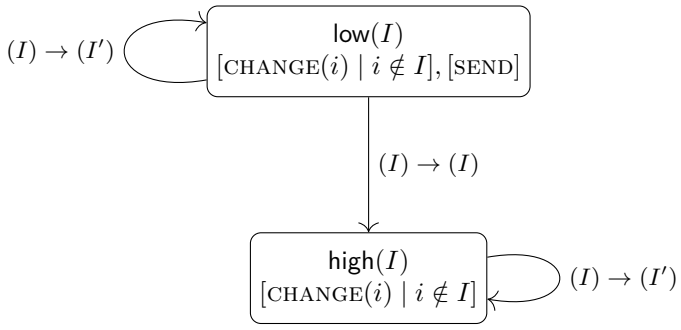
SPLIT
$\mathsf{recv}(\ell, P * Q) \Rrightarrow_\top \mathsf{recv}(\ell, P) * \mathsf{recv}(\ell, Q)$

WEAKEN
$(P \mathrel{-\!\!*} Q) \vdash \mathsf{recv}(\ell, P) \mathrel{-\!\!*} \mathsf{recv}(\ell, Q)$

2 Proof

We assume we picked some namespace $\mathcal{N}$. Since we only export triples at the full mask, the concrete choice does not matter.

We start by defining the core STS. Let BARRIER be the RA obtained from the following STS:



Here, $I \in \wp^{\text{fin}}(\mathbb{N})$ is a finite set of identifiers.

The interpretation of the STS, the abstract predicates, and other important assertions, are given as follows:

$$\begin{aligned}
\mathsf{BarrierInv}(\ell, P)(\mathsf{low}(I)) &\triangleq \ell \mapsto 0 * \mathsf{ress}(P, I)\\
\mathsf{BarrierInv}(\ell, _)(\mathsf{high}(I)) &\triangleq \ell \mapsto 1 * \mathsf{ress}(\mathsf{True}, I)\\
\mathsf{ress}(P, I) &\triangleq \exists Q : I \xrightarrow{\text{fin}} \mathsf{Prop}.\, \triangleright(P \mathrel{-\!\!*} \mathop{\Large *}_{i \in I} Q(i)) * \mathop{\Large *}_{i \in I} i \Rrightarrow Q(i)\\
\mathsf{BarrierCtx}^{\gamma}(\ell, P) &\triangleq \mathsf{StsCtx}^{\mathcal{N}, \gamma}(\text{BARRIER}, \mathsf{BarrierInv}(\ell, P))\\
\mathsf{send}(\ell, P) &\triangleq \exists \gamma.\, \mathsf{BarrierCtx}^{\gamma}(\ell, P) * \mathsf{StsSt}^{\gamma}(\mathsf{low}(\emptyset), [\text{SEND}])\\
\mathsf{recv}(\ell, R) &\triangleq \exists \gamma, P, Q, i.\, \mathsf{BarrierCtx}^{\gamma}(\ell, P) * \mathsf{StsSt}^{\gamma}(\mathsf{low}(\{i\}), [\text{CHANGE}(i)]) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)
\end{aligned}$$

Proof of ALLOC.

$\{\mathsf{True}\}_{\top}$
`ref`(0)
$\{\ell.\, \ell \mapsto 0\}$
Allocate a new saved proposition i with content P
$\{\ell \mapsto 0 * i \Rrightarrow P\}$
Allocate an instance γ of BARRIER with interpretation $\mathsf{BarrierInv}(\ell, P)$ in state $\mathsf{low}(\{i\})$
$\{\mathsf{BarrierCtx}^{\gamma}(\ell, P) * \mathsf{StsSt}^{\gamma}(\mathsf{low}(\{i\}), [\text{CHANGE}(i), \text{SEND}]) * i \Rrightarrow P\}$
$\{\mathsf{send}(\ell, P) * \mathsf{recv}(\ell, P)\}$

Proof of SIGNAL.

$\{\mathsf{send}(\ell, P) * P\}_{\top}$
Context: $\mathsf{StsCtx}^{\mathcal{N}, \gamma}(\text{BARRIER}, \mathsf{BarrierInv}(\ell, P))$
$\{\mathsf{StsSt}^{\gamma}(\mathsf{low}(\emptyset), [\text{SEND}]) * P\}$
Open STS $\mathcal{N}$
  Context: s, I, $s = \mathsf{low}(I)$
  $\{\triangleright\mathsf{BarrierInv}(\ell, P)(s) * P\}_{\top - \mathcal{N}}$
  $\{\ell \mapsto 0 * \triangleright\mathsf{ress}(P, I) * P\}$
  $\ell \leftarrow 1$
  $\{\ell \mapsto 1 * \mathsf{ress}(P, I) * P\}$
  $\left\{\ell \mapsto 1 * P * \triangleright(P \mathrel{-\!\!*} \mathop{\Large *}_{i \in I} Q(i)) * \mathop{\Large *}_{i \in I} i \Rrightarrow Q(i)\right\}$

$\left\{\ell \mapsto 1 * \left(\triangleright \mathop{\Large *}_{i \in I} Q(i)\right) * \mathop{\Large *}_{i \in I} i \Rrightarrow Q(i)\right\}$
$\{\ell \mapsto 1 * \mathsf{ress}(\mathsf{True}, I)\}$
$\{\mathsf{BarrierInv}(\ell, P)(\mathsf{high}(I))\}$
Context: $(s, [\textsc{send}]) \rightarrow^* \mathsf{high}(I)$
$\{\mathsf{True}\}_\top$

Proof of WAIT**.** The let-expansion of the code here is just performed to clarify the proof outline. The actual Coq proof works on the code as given above.

Context: $\{\mathsf{recv}(\ell, R)\}\ \texttt{wait}(\ell)\ \{R\}_\top$
$\{\mathsf{recv}(\ell, R)\}_\top$
Context: $\mathsf{StsCtx}^{\mathcal{N},\gamma}(\textsc{barrier}, \mathsf{BarrierInv}(\ell, P))$
$\{\mathsf{StsSt}^\gamma(\mathsf{low}(\{i\}), [\textsc{change}(i)]) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}$
Open STS $\mathcal{N}$
Context: s, I, $s \in \{\mathsf{low}(I), \mathsf{high}(I)\}$, $i \in I$
$\{\triangleright\mathsf{BarrierInv}(\ell, P)(s) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}_{\top - \mathcal{N}}$
Case $s = \mathsf{low}(I)$
$\{\ell \mapsto 0 * \triangleright\mathsf{ress}(P, I) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}$
$\texttt{let}\ y := !\ell\ \texttt{in}$
Pick $s' \triangleq \mathsf{low}(I)$, $T' \triangleq \{[\textsc{change}(i)]\}$, Context $(s, [\textsc{change}(i)]) \rightarrow^* (s', T')$
$\{y.\, y = 0 * \mathsf{BarrierInv}(\ell, P)(s') * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}$
Case $s = \mathsf{high}(I)$
$\{\ell \mapsto 1 * \triangleright\mathsf{ress}(\mathsf{True}, I) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}$
$\texttt{let}\ y := !x\ \texttt{in}$
$\{y.\, \ell \mapsto 1 * \mathsf{ress}(\mathsf{True}, I) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}$
Context: $y = 1$
$\left\{\ell \mapsto 1 * \left(\mathop{\Large *}_{j \in I} \exists Q'.\, j \Rrightarrow Q' * \triangleright Q'\right) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\right\}$
$\left\{\ell \mapsto 1 * \left(\mathop{\Large *}_{j \in I \setminus \{i\}} \exists Q'.\, j \Rrightarrow Q' * \triangleright Q'\right) * i \Rrightarrow Q' * \triangleright Q' * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\right\}$
$\{\ell \mapsto 1 * \mathsf{ress}(\mathsf{True}, I \setminus \{i\}) * \triangleright Q' * \triangleright Q = \triangleright Q' * \triangleright(Q \mathrel{-\!\!*} R)\}$
Pick $s' \triangleq \mathsf{high}(I \setminus \{i\})$, $T' \triangleq \emptyset$, Context $(s, [\textsc{change}(i)]) \rightarrow^* (s', T')$
$\{\mathsf{BarrierInv}(\ell, P)(s') * \triangleright R\}$
$\{\mathsf{BarrierInv}(\ell, P)(s') * (y = 0 * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R) \lor y = 1 * \triangleright R)\}$
$\{(y = 0 * \mathsf{StsSt}^\gamma(\mathsf{low}(\{i\}), [\textsc{change}(i)]) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)) \lor (y = 1 * \mathsf{StsSt}^\gamma(\mathsf{low}(\emptyset)) * \triangleright R)\}_\top$
$\texttt{if}\ y = 1\ \texttt{then}$
$\{R\}\ ()\ \{R\}$
$\texttt{else}$
$\{\mathsf{StsSt}^\gamma(\mathsf{low}(\{i\}), [\textsc{change}(i)]) * i \Rrightarrow Q * \triangleright(Q \mathrel{-\!\!*} R)\}$
$\{\mathsf{recv}(\ell, R)\}\ \texttt{wait}(\ell)\ \{R\}$

Proof of SPLIT**.**

$\{\mathsf{recv}(\ell, R * R')\}_\top$
Context: $\mathsf{StsCtx}^{\mathcal{N},\gamma}(\textsc{barrier}, \mathsf{BarrierInv}(\ell, P))$
$\{\mathsf{StsSt}^\gamma(\mathsf{low}(\{i\}), [\textsc{change}(i)]) * i \Rrightarrow P' * \triangleright(P' \mathrel{-\!\!*} R * R')\}$
Open STS $\mathcal{N}$
Context: s, I, $s \in \{\mathsf{low}(I), \mathsf{high}(I)\}$, $i \in I$
$\{\triangleright\mathsf{BarrierInv}(\ell, P)(s) * i \Rrightarrow P' * \triangleright(P' \mathrel{-\!\!*} R * R')\}_{\top - \mathcal{N}}$
In any case, we have $\ell \mapsto v_s$ for some v_s and $\triangleright\mathsf{ress}(P_s)$ for some P_s

$\{\ell \mapsto v_s * \triangleright \mathsf{ress}(P_s, I) * i \Rrightarrow P' * \triangleright(P' \mathrel{-\!\!*} R * R')\}$

$\left\{\ell \mapsto v_s * \triangleright\triangleright(P_s \mathrel{-\!\!*} \mathop{\Large *}_{j \in I} Q(j)) * i \Rrightarrow P' * \triangleright(P' \mathrel{-\!\!*} R * R') * \triangleright \mathop{\Large *}_{j \in I} j \Rrightarrow Q(j)\right\}$

$\left\{\ell \mapsto v_s * \triangleright \left(\triangleright(P_s \mathrel{-\!\!*} Q(i) * \mathop{\Large *}_{j \in I \setminus \{i\}} Q(j)) * \triangleright Q(i) = \triangleright P' * (P' \mathrel{-\!\!*} R * R') * \mathop{\Large *}_{j \in I \setminus \{i\}} j \Rrightarrow Q(j)\right)\right\}$

Allocate new saved propositions i', i'' not in I with contents R, R'

$\left\{\ell \mapsto v_s * i' \Rrightarrow R * i'' \Rrightarrow R' * \triangleright \left(\triangleright(P_s \mathrel{-\!\!*} R * R' * \mathop{\Large *}_{j \in I \setminus \{i\}} Q(j)) * \mathop{\Large *}_{j \in I \setminus \{i\}} j \Rrightarrow Q(j)\right)\right\}$

Pick $Q' \triangleq (Q \setminus \{i\})[i' \mapsto R, i'' \mapsto R']$, $I' \triangleq I \setminus \{i\} \cup \{i, i''\}$

$\left\{\ell \mapsto v_s * i' \Rrightarrow R * i'' \Rrightarrow R' * \triangleright \left(\triangleright(P_s \mathrel{-\!\!*} \mathop{\Large *}_{j \in I'} Q'(j)) * \mathop{\Large *}_{j \in I'} j \Rrightarrow Q'(j)\right)\right\}$

$\{\ell \mapsto v_s * \triangleright \mathsf{ress}(P_s, I') * i' \Rrightarrow R * i'' \Rrightarrow R'\}$

If $s = \mathsf{low}(I)$:

Pick $s' \triangleq \mathsf{low}(I')$, $T' \triangleq \{[\textsc{change}(i'), \textsc{change}(i'')]\}$, Context $(s, [\textsc{change}(i)]) \to^* (s', T')$

Otherwise, $s = \mathsf{high}(I)$:

Pick $s' \triangleq \mathsf{high}(I')$, $T' \triangleq \{[\textsc{change}(i'), \textsc{change}(i'')]\}$, Context $(s, [\textsc{change}(i)]) \to^* (s', T')$

$\{\triangleright \mathsf{BarrierInv}(\ell, P)(s') * i' \Rrightarrow R * i'' \Rrightarrow R'\}$

$\{\mathsf{StsSt}^\gamma(\mathsf{low}(\{i', i''\}), [\textsc{change}(i'), \textsc{change}(i'')]) * i' \Rrightarrow R * i'' \Rrightarrow R'\}$

$\{\mathsf{recv}(\ell, R) * \mathsf{recv}(\ell, R')\}$

Proof of WEAKEN**.**

$\{(R \mathrel{-\!\!*} R') * \mathsf{recv}(\ell, R)\}_\top$

Context: $\mathsf{StsCtx}^{\mathcal{N},\gamma}(\textsc{barrier}, \mathsf{BarrierInv}(\ell, P))$

$\{\mathsf{StsSt}^\gamma(\mathsf{low}(\{i\}), [\textsc{change}(i)]) * i \Rrightarrow P' * \triangleright(P' \mathrel{-\!\!*} R) * (R \mathrel{-\!\!*} R')\}$

$\{\mathsf{recv}(\ell, R')\}_\top$

3 Client

In Coq, we also verified safety of the following sample client, to demonstrate the higher-order nature of the specification. It shows that we can transfer Hoare triples through the barrier.

$$\begin{array}{l}\mathsf{client} \triangleq \texttt{let}\ y := \texttt{ref}(0)\ \texttt{in}\ \texttt{let}\ \mathsf{b} := \texttt{newbarrier}()\ \texttt{in} \\ \quad \begin{array}{l} y \leftarrow (\lambda z.\, z + 42); \\ \texttt{signal}(b) \end{array} \Bigg\| \begin{array}{l} \texttt{wait}(b); \\ (!y)(12) \end{array} \Bigg\| \begin{array}{l} \texttt{wait}(b); \\ (!y)(17) \end{array}\end{array}$$

The verification proceeds with the following resource being transferred through the barrier:

$$P_1 \triangleq \exists f.\, y \mapsto f * \forall n.\, \{\mathsf{True}\}\ f\ \{v.\, v = n + 42\}$$

To keep the code simpler and more to the point, this client does not actually perform any "useful work" in the waiting threads. Clearly, the closure stored in y satisfies the Hoare triple given above. Hence the first thread can produce P, and give it up when calling `signal`.

What about the two other threads? We have to find some way to split P in two pieces, such that each piece suffices to justify calling the contents of y. The key observation here is that, since both threads are only *reading* from y, we can *split* the $y \mapsto f$ into two pieces $y \overset{0.5}{\longmapsto} f$. Such a *fractional permission* for a heap location is enough to read from y, but not to write to it. So we define:

$$P_{0.5} \triangleq \exists f.\, y \overset{0.5}{\longmapsto} f * \forall n.\, \{\mathsf{True}\}\ f\ \{v.\, v = n + 42\}$$

In our logic, Hoare triples are *duplicable* assertions, since they do not assert exclusive ownership of any resource. Hence we can show that

$$P_1 \mathrel{-\!\!*} P_{0.5} * P_{0.5}$$

Combining this with the rule for splitting recv, we can equip *both* waiting threads with the permission $\mathsf{recv}(b, P_{0.5})$. After calling `wait`, they both have $P_{0.5}$, which justifies derferencing y and calling its contents with some numerical argument.

References

[1] Mike Dodds et al. "Verifying Custom Synchronization Constructs Using Higher-Order Separation Logic". In: *TOPLAS* 38.2 (2016), p. 4. DOI: 10.1145/2818638. URL: http://doi.acm.org/10.1145/2818638.